Sensitivity Analysis of the *i** Optimisation Model

Chitra M. Subramanian[1*], Aneesh Krishna[1*], Arshinder Kaur[2]

[1] Department of Computing, Faculty of Science and Engineering, Curtin University, Perth, Australia.
[2] Department of Management Studies, Indian Institute of Technology Madras, Tamil Nadu, India and School of Information Systems, Curtin Business School, Curtin University, Australia.

* Corresponding author. Email: chitra.muniyapp@postgrad.curtin.edu.au, A. Krishna@curtin.edu.au
Manuscript submitted August 10, 2015; accepted November 20, 2015.
doi: 10.17706/jsw.11.1.10-26

Abstract: Requirement elicitation is an important activity in early requirement engineering. Several well-known approaches and goal models have been developed to deal with requirement elicitation. The *i** goal model is used to represent the socio-technical domains and can be used for analysis in early requirement analysis. The elicitation process is complicated by the incomplete and imprecise input data available for analysis. Imprecise and unavailable data can be captured by fuzzy logic and then it can be managed by the popular operation research techniques known as optimisation. This paper presents a formal multi-objective optimisation model, for the *i** framework, with regard to requirement elicitation. The optimisation model has the capacity to handle large and complex systems. The optimisation model has been expanded to include sensitivity analysis, in order to facilitate useful information on input data for the requirement analyst. The proposed approach is explained using the London Ambulance System case study and is evaluated using a simulation based analysis.

Key words: Goal model, *i** framework, multi-objective optimisation, sensitivity analysis.

1. Introduction

Inadequate requirements are a major cause of any software system's failure. Requirements Engineering (RE) is a branch of software engineering which involves discovering, documenting and managing the requirements of any computer-based system [1]. Requirements elicitation, modelling and analysing requirements, and communicating requirements are some of the major activities of RE. The first and most crucial step in the RE process is elicitation. Elicitation identifies the stakeholders as well as the goals/tasks of the system. The goals/tasks represent the system objectives which must be met. A goal can either be a functional (behavioural) goal or non- functional (soft) goal. These two types of goals define the utility of a software system. A non-functional requirement affects either a single functional requirement or the system as a whole [2]. Therefore, to build a better software system the analyst performs the analysis of goals during the early stages of RE.

Goal-Oriented Requirement Engineering (GORE) is a modelling approach that models the requirements of a given software system in terms of goals. Some well-known GORE frameworks are the Non-Functional Requirements (NFR) framework [3], the Knowledge Acquisition in Automated Space (KAOS) [4], the *i** framework [5], the Tropos [6], the Goal-Oriented Requirement Language (GRL) [7] and the Attributed Goal-Oriented Requirements Analysis (AGORA) [8].

Among the different goal-oriented models, the *i** framework captures the social elements of the system and can be used for reasoning, especially at the requirements level [9]. Apart from modelling, goal models are used to evaluate design alternatives in which different design options are explored and the best ones are selected. For alternative design evaluation many approaches, both quantitative [7], [10], and qualitative [9], [11], have been proposed in the RE literature. For the *i** framework only qualitative analysis [9] has been proposed.

Previous works suggested the idea of enhancing the *i** framework to support quantitative reasoning; including the inter-actor dependencies [12], [13]. Though quantitative reasoning has been increasingly accepted, the optimisation models have attracted significantly less consideration. In practice, the input data required for evaluation are incomplete, or unobtainable, or imprecise. Moreover, when presented with real-life RE problems one usually has to deal with multiple goals, of which each may be significantly important to address in relation to the RE problem presented. The priority of these goals may be different but are important to consider at the same time. These goals can be conflicting or congruent. There is a need to address such issues with a method which considers multi-objective optimisation.

Furthermore, Scalability is another issue associated with requirement evaluation. It is difficult to assign values to the goals of the goal models in a large and complex system. Hence the decision making becomes a crucial task [5], [14]-[16]. Optimisation of an operation research technique is a method used to obtain a best possible solution [17], [18]. Operation research is generally used as a technique to maximise or minimise parameters, like profit and cost, within system considerations. Therefore, optimisation can be applied to the *i** framework to address the scalability problem.

Another interesting point of optimisation is sensitivity analysis. Sensitivity analysis is employed to detect the system's behaviour when input data changes. The main advantage of this technique is that a thorough investigation of the estimation of the input variables is made before making a final decision. It also aids in identifying the errors in the model and comprehending the effect of input parameters. The concept of sensitivity analysis is new to software engineering domains and this work is the earliest attempt at applying sensitivity analysis in case of the *i** framework. The only other research work on sensitivity analysis has been conducted by Affleck *et al*. [19]. However, this research was conducted on a different framework known as NFR.

This paper presents a complete optimisation model for the *i** framework to analyse the impact that alternative options have on the soft goals and for deciding on the optimal design options amongst the alternatives. Given an *i** goal model, the multi-objective functions representing the behaviour of that particular design are obtained. The solutions to the objective functions are computed and are used in goal analysis. The model is validated through simulation-based analysis. Furthermore, this papers examines the application of sensitivity analysis to the *i** framework in order to provide the analyst with extra information about the quantitative values selected.

The remainder of the paper is structured as follows: Section 2 explains goal-oriented requirement analysis of the *i** framework, quantitative analysis of the *i** framework , goal analysis using optimisation and application of optimisation to case studies; Section 3 presents sensitivity analysis; Section 4 discusses the related works; and Section 5 concludes the paper.

2. Background

2.1. Goal-Oriented Requirement Analysis Using the *i** Framework

The *i** framework introduced by Eric Yu [5], [20], models social elements of a system and can be used in the early requirements analysis stages. The Strategic Dependency (SD) model and the Strategic Rationale (SR) model are the two types of diagrams which are employed in modelling. The Strategic Dependency diagram represents a stakeholder's relationships while the Strategic Rationale diagram represents the internal and intentional relationship of each stakeholder.

An SD model is a graph in which the nodes represent the actors and the links represent the interdependency between the actors. Goal, soft goal, task and resources are the intentional elements. A dependency can be any one of the intentional elements. An SD model is a higher level of abstraction representing the actors' dependency upon each other. An SD model targets external relationships and does not disclose details of internal structure. An SR model assigns the intentional elements, goals, tasks, resources and soft goals to the actors. It describes how the actors achieve their goals. Intentional elements are linked by MEANS-END relationships, TASK decomposition and soft goal contributions. An SR model can be viewed as a graph that shows the decomposition of high-level goals into lower level goals by the MEANS-END / TASK decomposition. In the MEANS-END relationships a mean node can represent a soft goal or a task, and an end node can be a goal, a soft goal, a resource or a task. A MEANS-END links a task to a goal implying that a particular method is used to achieve a goal. TASK decomposition shows the sub-goals, resources and soft goals that are to be carried out to ensure the success of a task. A soft goal contribution can be any of the following types: help, make, some+, some-, hurt, or break. An example of the SR model is shown in Fig. 1. This figure shows graphical representation of the partial *i** model [21] from the London Ambulance Service [LAS]. The LAS is a computer-aided system used to automate the dispatch of the ambulance to the emergency scene with an arrangement that the ambulance be dispatched in 3 seconds and arrive at the scene in 11 seconds. However, the system failed to address the time requisite and was also unreliable and crashed. The LAS is used as a standard case study for goal modelling by the RE research community [21]. In this paper, we also use the LAS case study for optimal inter-actor dependency goal analysis. This partial SR diagram depicts four actors known as the *Ambulance Crew, the Resource Allocator, the Resource Allocator Module* and the *Human Resource Allocator,* as well as some of their intentional relationships. The *Resource Allocator* actor has the goal *Becollected[IncInfo]* which represents the incident information that is to be collected. This goal can be accomplished in two ways using either paper-based information or network-based information. So the goal *Becollected[IncInfo]* is decomposed into two tasks known as the *ByPaperbased Form* and the *ByDatabase or Network.* The goal *Becollected[IncInfo]* represents a decision point and similarly the goal *BeGenerated[MobileInst]* represents which vehicle is to be assigned to the incident scene. This information can be generated either by a computer-based algorithm or by a human. So this goal is again *OR* decomposed into two tasks namely the *By Machine Based Algorithm* and the *By Human Decision* and hence the goal *BeGenerated[MobileInst]* becomes the decision point. The selection of a task for these goals *Becollected[IncInfo]* and *Be Generated [MobileInst*] influences the satisfaction levels of non-functional goals or soft goals namely *Timeliness [mobilization]* and *Optimal[mobInst]*.

There are inter-actor dependencies between the actors. These inter-actor dependencies demonstrate that an actor depends on another actor for its goal accomplishment. The actor *Ambulance Crew* depends on the actor *Resource Allocator* through the dependency *Optimal [mobInst]*. The soft goal known as the *Quality[service]* of the actor *Ambulance Crew* depends upon the accuracy of the optimal information that has been collected. Additionally, the actor *Resource Allocator's* goal *BeGenerated[MobileInst]* depends upon the *ResourceAllocatorModule's* task *ByMachineBasedAlgorithm* and the actor *HumanResourceAllocator's* task *ByHumanDecision*. These inter-actor dependencies also influence the decision making of each actor.

2.2. Quantitative Analysis of *i** Framework

In Fig. 1 the actor *Resource Allocator* has two decision points, namely the goals *Becollected[IncInfo]* and *BeGenerated[MobileInst].* These two goals are OR decomposed into two tasks. The requirement analyst has to select an alternative task. The selection of the task also contributes to the non-functional requirements (represented by soft goals) of the goal model. Hence analysts encounter the problem of selecting an alternative task that maximises the satisfaction levels of the soft goals. For example, in this case study, LAS the analyst has to make a selection from the alternative tasks *Bypaper based form* or *By Database or Network in order* to maximise the satisfaction level of the top soft goals known as *Timeliness[mobilization]*, *optimal[mobInst]* of the actor *Resource Allocator.* The *Timeliness[service]* of the actor *Ambulance Crew* depends upon the *Optimal[MobInst]* of the actor known as the *Resource Allocator.* This scenario continues with the selection of the alternative options for the actor *Resource Allocator* affecting the soft goals of the actor known as the *Ambulance Crew.*

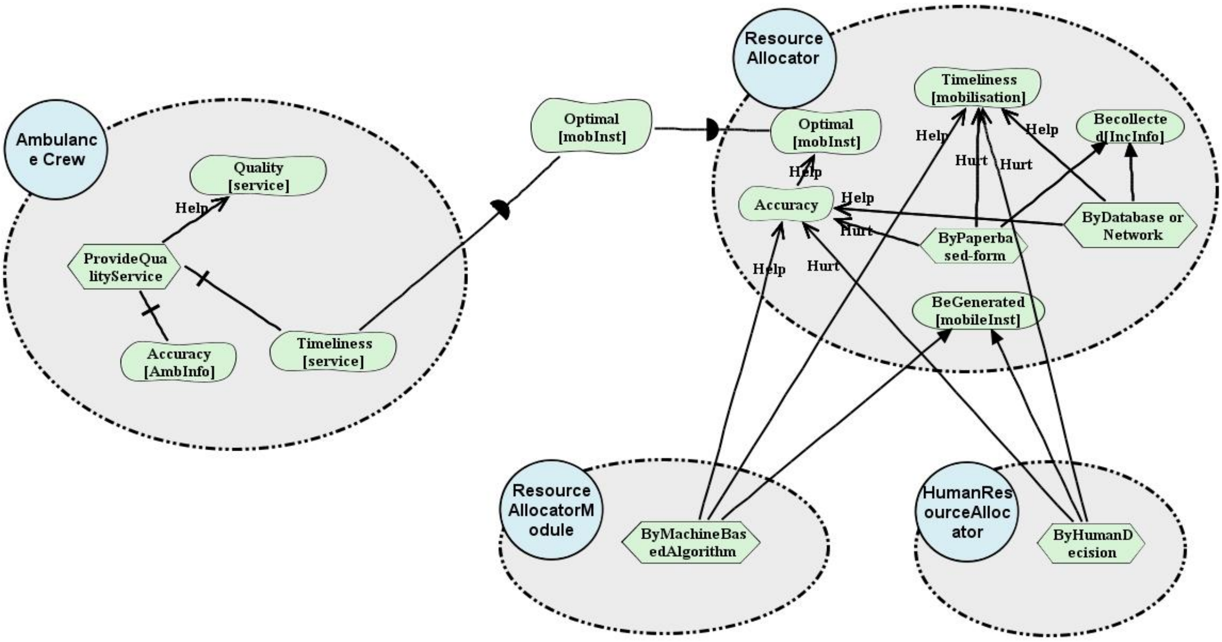

Fig. 1. SR diagram for LAS (adapted from [21]).

We have developed a quantitative reasoning for alternative choices based on inter-actor dependencies for the *i** framework [12], [13]. In this framework, the requirement analyst assigns weights to the soft leaf goals in percentages from 0 to 100. Next, the impacts of the goals or tasks are denoted by fuzzy numbers because assigning impacts can lead to imprecision due many analysts assigning different values and sometimes they are subjective. Therefore, it is easy to give a judgement within a range which can be defined by a fuzzy number rather than giving one numerical value. The impacts of the goals or task to soft goals known as *make, help, some+, some-, hurt,* and *break,* are represented by triangular fuzzy numbers. As an illustration, the impacts are represented by triangular fuzzy numbers (0.64, 0.80, 1), (0.48, 0.64, 0.80), (0.32, 0.48, 0.64), (0.16, 0.32, 0.48), (0, 0.16, 0.32) and (0, 0, 0.16) correspondingly. The correlation links are represented by fuzzy numbers because these impacts have a direct influence on the degree of the satisfaction of the soft goals and are used to avoid the imprecision associated with decision making. Once this data is collected, leaf soft goal scores are calculated for each alternative option. These scores are then propagated to the soft goals that are higher in the hierarchy. In the calculation of the scores of soft goals any dependencies are also taken into consideration. An actor depends on one or more actors for its goal achievement. Goals have to be analysed by considering the dependencies amongst the actors. The top soft goals' (goals that are top in the hierarchy) scores are compared in order to select the alternative option that best satisfies the top soft goals.

The inter-actor quantitative analysis is explained with LAS as a running example. Let us assume that an analyst assigns the weights 70%, 60%, 70% and 50% to the leaf soft goals (LSG) *Accuracy[AmbInfo]*, *Timeliness[service]*, *Accuracy* and *Timeliness[mobilization]* respectively and it is represented by ω_L. The goal

Becollected[IncInfo] of the actor *Resource Allocator* has two tasks, namely *ByPaperbased-form* and *ByDatabase* or *Network*. The analyst selects the first option known as *ByPaperbased-form* and performs the goal analysis to find the impact the selection of this option. Next the impacts of this alternative to the leaf soft goals are determined in the form of triangular fuzzy numbers. It is referred to as $\bar{C}_{A\times L}$ where A is an alternative option that is selected and L is a leaf soft goal. The impact of the alternative *ByPaperbased-form* to the leaf soft goals Accuarcy*[AmbInfo]*, *Timeliness[service]*, *Accuracy* and *Timeliness[mobilization]* are (0.48, 0.64, 0.80), (0.48, 0.64, 0.80) , (0, 0.16, 0.32) and (0, 0.16, 0.32) accordingly. An LSG score is calculated using its weight and impact for the selected alternative option. The leaf soft goal score is referred to as $\bar{S}_{\mathrm{L}}$. An actor may depend on another actor for its goal performance. The inter-actor dependencies may influence the decision making of the alternative options. The dependency link is considered to be the 'MAKE' contribution. If the dependency score and dependency impact are denoted by $\bar{S}_d$ and $\bar{I}_d$ correspondingly and if there are '*nd*' dependencies, then the equation for score calculation of an i^{th} leaf soft goal for j^{th} alternative of k^{th} actor is given by the Equation 1 below:

$$\bar{S}_{Lijkt} \;=\; \bar{C}_{(Aj*Li)k} \times \omega_{Lik} + \sum_{b=1}^{nd} \left(\bar{S}_{db} \times \bar{I}_{db}\right) \tag{1}$$

where t is the hierarchy level, and for leaf soft goals t is zero.

Thus using (1) the calculated scores of the LSGs *Accuarcy[AmbInfo]*, *Timeliness[service]*, *Accuracy* and *Timeliness[mobilization]* are (0.336, 0.448, 0.56), (0.288, 0.455, 0.65), (0, 0.11,0 .224) and (0, 0.08, 0.16) respectively. Next, the LSG scores are propagated backwards in the goal hierarchy until the top soft goals so as to find the scores of the soft goals. The soft goal (SG) score is referred to as $\bar{S}_{SG}$. The score calculation of an i^{th} soft goal for j^{th} alternative of k^{th} actor at t^{th} level in the hierarchy is given by Equation 2 below:

$$\bar{S}_{SGijkt} = \sum_{d=1}^{nc}((\bar{C}_{SGi\times(SGd\,|\,LSGd)})\times(\bar{S}_{Ldjk(t-1)}|\bar{S}_{SGdjk(t-1)})) + \sum_{b=1}^{nd}\left(\bar{S}_{d'b}\times\bar{I}_{d'b}\right) \tag{2}$$

where $\bar{C}_{\mathrm{SGi}\times(SGd\,|\,LSGd)}$ is the correlation link between the i^{th} soft goal and its d^{th} child which may be a soft goal or a leaf soft goal, $(\bar{S}_{Ldjk(t-1)}$ is the score of its d^{th} leaf soft goal child, $\bar{S}_{SGdjk(t-1)}$ is the score of its d^{th} soft goal child, | represents *or*, $\bar{S}_{d'b}$ is the score of its b^{th} dependent, $\bar{I}_{d'b}$ is the b^{th} dependent impact, '*nc*' is the number of its children and '*nd*' is the number of dependencies. By using (2), the calculated scores of the top soft goals *Quality*[*service*] and *Optimal[mobInst]* are (0.29, 0.515, 0.81) and (0, 0.07, 0.1792). These scores are defuzzified so as to obtain a quantifiable value. It shows the degree of satisfaction of the top soft goals for the selected alternative option. The defuzzified scores are 100% and 16% for *Quality[service]* and *Optimal[mobInst]* respectively. Similarly, the analyst has to perform the analysis to find the satisfaction values for the alternative options, known as *ByDatabase or Network*. By performing this analysis, the defuzzified scores of the top soft goals *Quality[service]* and *Optimal[mobInst],* are 100% and 59% accordingly. By comparing the scores of the two alternatives, the option *ByDatabase or Network* is found to better satisfy the soft goals. Hence the analyst decides to select the option, *ByDatabase or Network.*

Since the weights are subjective to the analyst, different scores are obtained for the same soft goals based on the analyst's weights. To illustrate this, let us assume that if another analyst assigns the weights 50%, 50%, 60% and 40%, to the leaf soft goals known as *Accuarcy*[*AmbInfo*], *Timeliness* [*service*], *Accuracy* and *Timeliness[mobilization]* respectively. The calculated scores of the top soft goals *Quality*[*service*]and *Optimal* [*mobInst*] are now found to be 84% and 13% for the first alternative option *By Paperbased-form and* 84% and 50% for the second alternative option *ByDatabase or Network*. In the above analysis we can see that different scores are calculated for the same alternative with different weights to the leaf soft goals. Hence the scores of the soft goals are subjective depending upon the subjective selection of weights made by the requirement analyst. So as to avoid these subjective scores, it is proposed that an optimisation model is used to find the weights of the leaf soft goals. To determine the optimal weight, we use multi-objective optimisation.

2.3. Multi-objective Optimisation

Nowadays in all professions, optimisation is used as a technique for decision making. Optimisation is a method of selecting the best or optimal alternatives from a list of possible choices [22]. Linear programming, non-linear programming and quadratic programming are some of the optimisation research techniques used.

For real-world problems, the single objective optimisation technique is too inadequate to identify the solution. Therefore, the techniques for solving problems with multiple objective optimisation have been developed [23]. In the LAS case study (Fig. 1), the goal *Becollected* [*IncInfo*] has two different choices namely *ByPaperbased-form* and *ByDatabase or Network* for the actor *Resource Allocator*. Now the task of the requirement analyst is to select the best or optimal alternative among these two choices. Each choice is considered to be an objective and hence this problem can be solved by using multi-objective optimisation.

A multi-objective optimisation problem is written mathematically as:

$$\underset{x\,\in\, Y}{Max/Min}\ [f1\ (x), f2\ (x), \ldots., fn(x)] \tag{3}$$

where $f1$, $f2,\ldots.,$ fn are scalar functions and Y is the set of constraints. A multi-objective optimisation generates a set of solutions which are called Pareto solutions or a Pareto frontier. The best solution is selected from a Pareto frontier.

2.3.1. Optimal *i** framework

As this approach aims to completely automate the analysis process, there is a need to minimise the analyst's involvement in assigning the weights to leaf soft goals. Assigning weight in the case of a large goal model, and also the preferences for weight, may vary from analyst to analyst. So, to automate and handle the scalability issue, it is proposed that the multi-objective optimisation is applied.

Optimisation is performed so as to find the weights of the leaf soft goals and thereby identify an alternative option by which the soft goals satisfaction can be maximized. This minimizes the analyst's interaction and also, by automating the process, it can handle large complex systems.

To model the optimisation, the SR diagram is viewed as the directed graph G (N, A) where N is the set of nodes and A is the set of arcs. The intentional elements of soft goals, goals, and tasks in the SR diagram are assumed to be the nodes of the directed graph G and the means-end, task-decomposition and operational contribution links are assumed to be the arcs of the graph G.

An objective function for the optimisation model is formed from the calculations of variables of the graph. For any node that is a leaf soft goal there is a weight represented by ω_{Lik}, meaning that the weight of the i^{th} leaf soft goal of the k^{th} actor. Additionally, any arc from a goal/task to a leaf soft goal that is an impact of goal/task is denoted by a triangular fuzzy number $\bar{C}_{Aj\times Li,}$ denoting the impact of the j^{th} alternative option on i^{th} leaf soft goal. The i^{th} leaf soft goal score is calculated from the weight of the i^{th} leaf soft goal and its impact for j^{th} alternative and is represented by $\bar{S}_{Lij}$.

The score of i^{th} leaf soft goal for j^{th} alternative of k^{th} actor is given by Equation 1 as :

$$\bar{S}_{Lijk} \;=\; \bar{C}_{(Aj\times Li)k} \times \omega_{Lik} \;+\; \sum_{b=1}^{nd} (\bar{S}_{db} \times\; \bar{I}_{db}\,)$$

Since the hierarchy level (t) for leaf soft goals is zero, we avoid 't' in the leaf soft goal score representation. To get maximum satisfaction for any top soft goal, the sum of the leaf soft goal scores for an alternative has to be maximized. Hence the objective function for the first alternative of k^{th} actor with 'm' leaf soft goals is represented by:

$$\text{Max}\,(\bar{S}_{L11k} + \bar{S}_{L21k} + \ldots\ldots\ldots + \bar{S}_{Lm1k}) \quad (4)$$

Using Equation 1 of leaf soft goal score calculation, Equation 4 is expanded as below

$$\text{Max}\{\bar{C}_{(A1\times L1)k} \times \omega_{L1k} + \sum_{b=1}^{nd_1} (\bar{S}_{db} \times \bar{I}_{db}) + \bar{C}_{(A1\times L2)k} \times \omega_{L2k} + \sum_{b=1}^{nd_2} (\bar{S}_{db} \times \bar{I}_{db}) + \bar{C}_{(A1\times Lm)k} \times \omega_{Lmk} + \sum_{b=1}^{nd_m} (\bar{S}_{db} \times \bar{I}_{db})\} \quad (5)$$

and equation (5) can be rewritten as given below.

$$\text{Max}\{\bar{C}_{(A1*L1)k} \times \omega_{L1k} + \bar{C}_{(A1\times L2)k} \times \omega_{L2k} + \ldots + \bar{C}_{(A1\times Lm)k} \times \omega_{Lmk} + \sum_{b=1}^{nd_1} (\bar{S}_{db} \times \bar{I}_{db}) + \sum_{b=1}^{nd_2} (\bar{S}_{db} \times \bar{I}_{db}) + \ldots + \sum_{b=1}^{nd_m} (\bar{S}_{db} \times \bar{I}_{db})\}$$

$$\text{Max}\{\sum_{i=1}^{m} \bar{C}_{(A1\times Li)k} \times \omega_{Lik} + \sum_{i=1}^{m} \sum_{b=1}^{nd_i} (\bar{S}_{db} \times \bar{I}_{db})\} \quad (6)$$

The Equation 6 represents the object function for an i^* framework taking into consideration both the Strategic Dependency (SD) and Strategic Rationale (SR) model. To avoid complexity in solving the objective function, we are only optimizing SR without considering SD. The output of the optimisation model of SR is considered in simulation and further optimisation is done using SD. Therefore, the objective function for an i^* framework by considering only SR is given by Equation 6 as below:

$$\text{Max}\,\{\sum_{i=1}^{m} \bar{C}_{(A1\times Li)k} \times \omega_{Lik}\} \quad (7)$$

Subject to:

$$\omega_{L11}, \omega_{L21}, \ldots\ldots, \omega_{Lm1} \geq 0$$
$$\omega_{L11}, \omega_{L21}, \ldots\ldots, \omega_{Lm1} \leq 100$$

Supposing there are 'n' alternatives for a given k^{th} actor, then there are 'n' objective functions given by:

$$\left.\begin{aligned} F1(\omega_L) &= Max \sum_{i=1}^{m} \bar{C}_{(A1\times Li)k} \times \omega_{Lik} \\ F2(\omega_L) &= Max \sum_{i=1}^{m} \bar{C}_{(A2\times Li)k} \times \omega_{Lik} \\ &\ldots\ldots \\ &\ldots\ldots \\ Fn(\omega_L) &= Max \sum_{i=1}^{m} \bar{C}_{(An\times Li)k} \times \omega_{Lik} \end{aligned}\right\} \quad (8)$$

Subject to:

$$0 \leq \omega_{Lik} \leq 100 \quad \text{for } i = 1 \text{ to } m$$

Similarly, the objective functions are carried out for each actor in the SR model. In the case of a goal model in which the alternatives are the same in all actors, then a cumulative objective function involving all the actors can be used. Supposing a goal model has 'p' number of actors, the objective function for a j^{th} alternative option in the goal model is given by:

$$Fj(\omega_L) = Max\,(\sum_{i=1}^{m} \bar{C}_{(Aj\times Li)1} \times \omega_{Li1} + \sum_{i=1}^{m} \bar{C}_{(Aj\times Li)2} \times \omega_{Li2} + \ldots + \ldots + \sum_{i=1}^{m} \bar{C}_{(Aj\times Li)p} \times \omega_{Lip}) \qquad (9)$$

In short the function is given by:

$$Fj(\omega_L) = Max \sum_{k=1}^{p} \sum_{i=1}^{m} \bar{C}_{(Aj\times Li)k} \times \omega_{Lik} \qquad (10)$$

Therefore the objective functions for a goal model with '*n*' number of alternatives are given by:

$$\left.\begin{array}{c} F1(\omega_L) = Max \sum_{k=1}^{p} \sum_{i=1}^{m} \bar{C}_{(Aj\times Li)k} \times \omega_{Lik} \\ F2(\omega_L) = Max \sum_{k=1}^{p} \sum_{i=1}^{m} \bar{C}_{(Aj\times Li)k} \times \omega_{Lik} \\ \ldots\ldots \\ \ldots\ldots \\ Fn(\omega_L) = Max \sum_{k=1}^{p} \sum_{i=1}^{m} \bar{C}_{(Aj\times Li)k} \times \omega_{Lik} \end{array}\right\} \qquad (11)$$

Subject to:

$$0 \le \omega_{Lik} \le 100 \quad \text{for } i = 1 \text{ to } m \text{ and } k = 1 \text{ to } p$$
$$0 \le \bar{C}_{Aj\times Lik} \times \omega_{Lik} \le 1 \text{ for } i = 1 \text{ to } m, j = 1 \text{ to } n \text{ and } k = 1 \text{ to } p$$

In general, the objective functions are given by following equation:

$$Max\,[F1(\omega_L), F2(\omega_L), \ldots, Fn(\omega_L)] \qquad (12)$$
$$with\ \omega_L \in Y$$

where $n > 1$ and Y is the set of constraints defined.

These objective functions can be solved by using the scalarization or the weighted sum technique [18]. The new optimisation problem with unique objective function in the scalarization method is given by:

$$\begin{array}{c} \text{Max} \sum_{i=1}^{n} \gamma_i F_i(\omega_L) \\ \sum_{i=1}^{n} \gamma_i = 1 \\ 0 \le \omega_L \le 100 \\ \gamma_i \ge 0,\ i=1,2,\ldots, n \end{array} \qquad (13)$$

where γ denotes the weight associated with each objective function.

2.3.2. Multi-objective optimisation algorithms

Solving multi-objective optimisation problems is not as simple as for a conventional single-objective optimisation problem as there are multiple Pareto optimal solutions. On exploring different ways to solve multi-objective optimisation problems, one approach is to transform transforming the multi-objective optimisation problem into a unique objective optimisation problem. It is called scalarization or the weighted sum technique [23].

A multi-objective optimisation generates a set of solutions called a Pareto solutions or a Pareto frontier. The best solution is selected from a Pareto frontier. Evolutionary algorithms are prominent approaches for generating Pareto optimal solutions. Non-dominated Sorting Genetic Algorithm- II (NSGA- II) and Strength Pareto Evolutionary Algorithm 2(SPEA – 2) are standard evolutionary approaches. We have used NSGA-II [24] evolutionary algorithm that has an implementation in Matlab's Global Optimization Toolbox. We briefly explain the use such algorithm for finding optimal weights of the leaf soft goals in goal models.

The evolutionary algorithm begins from a population of randomly generated individuals and finds a solution over a number of iterations. The population in each iteration is known as generations. In each generation the fitness of the selected individual known as a chromosome (weight of leaf soft goal) in the population is evaluated. The fitness is the value of the objective function in the optimisation problem. The selected individuals from the current population are randomly mutated by a process called crossover to form a new generation, which is used in the next iteration. The algorithm can be terminated when either a maximum number of generations has been produced or a satisfactory fitness level has been reached for the population.

2.3.3. Encoding the optimisation problem for weights of the leaf soft goals

Binary representations are used to show the chromosomes in Genetic algorithms. Therefore, we define a mapping from weights of the leaf soft goals in the goal model to a binary representation. The aim of the objective function is to find the weights of the leaf soft goals. So the number of bits in the chromosome depends on the number of leaf soft goals in the goal model. In the LAS goal model the number of leaf soft goals is two and hence the number of bits in the chromosome is 2. Let us represent the actors *Ambulance Crew*, *Resource Allocator*, *Resource Allocator Module* and *Human Resource Allocator* as 1, 2, 3, and 4 respectively. The actor *Ambulance Crew* has no alternatives and has no objective functions. Then objective functions for the LAS are given by:

F1 = Max $(0.16\times \omega_{L12} + 0.16\times\omega_{L22})$ {Objective function for the alternative *ByPaperbased Form]*}
F2=Max $(0.64\times\omega_{L12} + 0.64\times\omega_{L22})$ {Objective function for the alternative *ByDatabase or Network]*}
F3=Max $(0.64\times\omega_{L13} + 0.64 \times\omega_{L23})$ {Objective function for the alternative *ByMachineBasedAlgorithm]*}
F4=Max $(0.16\times\omega_{L14} + 0.16 \times\omega_{L24})$ {Objective function for the alternative *ByHumanDecision]*}

Subject to
$0 \le \omega_{L\,i} \le 100$ for i = 1 to 2
Using scalarization method, the objective function is given by

$$\text{Max}\ \{\gamma_1 (0.16\times\omega_{L12} + 0.16 \times \omega_{L22}) + \gamma_2 (0.64\times\omega_{L12} + 0.64 \times \omega_{L22}) + \gamma_3 (0.64\times\omega_{L13}+ 0.64\times\omega_{L23}) + \gamma_4 (0.16\times\omega_{L14} + 0.16\times\omega_{L24})\}$$

$$\sum_{i=1}^{4} \gamma_i = 1$$

$$0 \le \omega_{L1j}, \omega_{L2j} \le 100, j = 2 \text{ to } 4$$

$$\gamma_i \ge 0, i = 1,2,3,4$$

The Fig. 2 illustrates the process of crossover with γ_1 =1 and γ_2, γ_3, γ_4 as zeros.

	ω_{L11}	ω_{L21}	ω_{L11}	ω_{L21}	
Population 1	23	56	0010111	0111000	F(ω)=12.64
Population 2	60	35	0111100	0100011	F(ω) =15.2
Crossover	60	56	0111100	0111000	F(ω)=18.29
	Abstract encoding		Algorithmic encoding		

Fig. 2. Crossover for optimal weights.

The process is repeated for a specified number of iterations or until an optimal solution is found.

2.4. Application of Optimisation to Case Study

In LAS case study, the actor the *Resource Allocator* has goals *Becollected[IncInfo]* and *BeGenerated[MobileInst]* as two decision points. The goal *Becollected[IncInfo]* is *OR* decomposed into two tasks

- *ByPaperbased Form* and
- *ByDatabase or Network*

and the goal BeGenerated*[MobileInst] OR* decomposed into two tasks

- *ByMachineBasedAlgorithm*
- *ByHumanDecision.*

The objective of the optimisation approach is to select an alternative for these two goals that achieves maximum satisfactions for the soft goals *Optimal[mobInst]* and *Timelineness[mobilization]* of the actor *Resource Allocator* and *Quality[service]* of the actor *Ambulance Crew*.

Let us represent the weight obtained from the optimiser as the optimal weight, which is denoted by $O\omega$. The leaf soft goal score (1) is updated as:

$$\bar{S}_{Lijkt} = \bar{C}_{(Aj \times Li)k} \times O\omega_{Lik} + \sum_{b=1}^{nd} (\bar{S}_{db} \times \bar{I}_{db}) \quad (1')$$

The LAS goal model consists of four alternatives known as *ByPaperbased Form*, *ByDatabase or Network, ByMachineBasedAlgorithm* and *ByHumanDecision*. Based on the number of alternatives, this model has four objective functions, one for each alternative. The objective function for the alternative *ByPaperbased Form,* denoted by F1 is given by:

$$F1\ (\omega_L) = \mathrm{Max}\sum_{i=1}^{2}(\bar{C}_{A1 \times Li2} \times \omega_{Li2})$$
$$F1 = Max\ (0.16 \times \omega_{12} + 0.16 \times \omega_{22})$$

where ω_{12}, ω_{22} represent the weights of the leaf soft goals *Accuracy and Timeliness* of actor *Resource Allocator(*Actor number 2*)*. For the convenience of solving the objective functions the defuzzified values of correlation links are used in the functions.

Similarly, the objective function for the other three options *ByDatabase or Network, ByMachineBasedAlgorithm* and *ByHumanDecision* are given by F2, F3 and F4 correspondingly:

$$F2 = Max\ (0.64 \times \omega_{12} + 0.64 \times \omega_{22})$$
$$F3 = Max\ (0.64 \times \omega_{13} + 0.64 \times \omega_{23})$$
$$F4 = Max\ (0.16 \times \omega_{14} + 0.16 \times \omega_{24})$$

The multi-objective functions for the LAS goal model are:

$$F1 = Max\ (0.16 \times \omega_{12} + 0.16 \times \omega_{22})$$
$$F2 = Max\ (0.64 \times \omega_{12} + 0.64 \times \omega_{22})$$
$$F3 = Max\ (0.64 \times \omega_{13} + 0.64 \times \omega_{23})$$
$$F4 = Max\ (0.16 \times \omega_{14} + 0.16 \times \omega_{24})$$

Subject to:

$$0 \le \omega_{ij} \le 100 \quad \text{for } i = 1 \text{ to } 2 \text{ and } j = 2 \text{ to } 4$$
$$0.64 \times \omega_{ij} \le 100 \text{ for } i = 1, 2 \text{ and } j = 2 \text{ to } 4$$
$$0.16 \times \omega_{ij} \le 100 \text{ for } i = 1, 2 \text{ and } j = 2 \text{ to } 4$$

By solving the above multi-objective functions using the MATLAB Genetic Algorithm, the weights of the leaf soft goals are identified and are presented in Table 1. These weights are now used in Equation (1') and thereby to find the optimal satisfaction of the soft goals and top soft goals. The calculated scores of the soft goals for the alternative option *ByDatabase or Network* are provided in Fig.3.

Fig. 3 shows that the alternative option *ByDatabase or Network* was estimated to achieve the top soft goals of *Quality[service](Ambulance Crew)*, *Optimal[mobInst](Resource Allocator)* and *Timeliness[mobilization](Resource Allocator*) in 100%, 76% and 100% of the cases correspondingly. Similarly by performing an analysis for other alternative options, the alternative options *ByDatabase or Network* and *ByMachineBasedAlgorithm* were found to be the optimal alternative options (Due to space restriction other alternative option values are not shown).

3. Sensitivity Analysis

Table 1. Weights of the Leaf Soft Goal

Actor	Leaf Soft Goal	Weight
Resource Allocator	Accuracy	0.9
	Timeliness[mobilization]	0.8

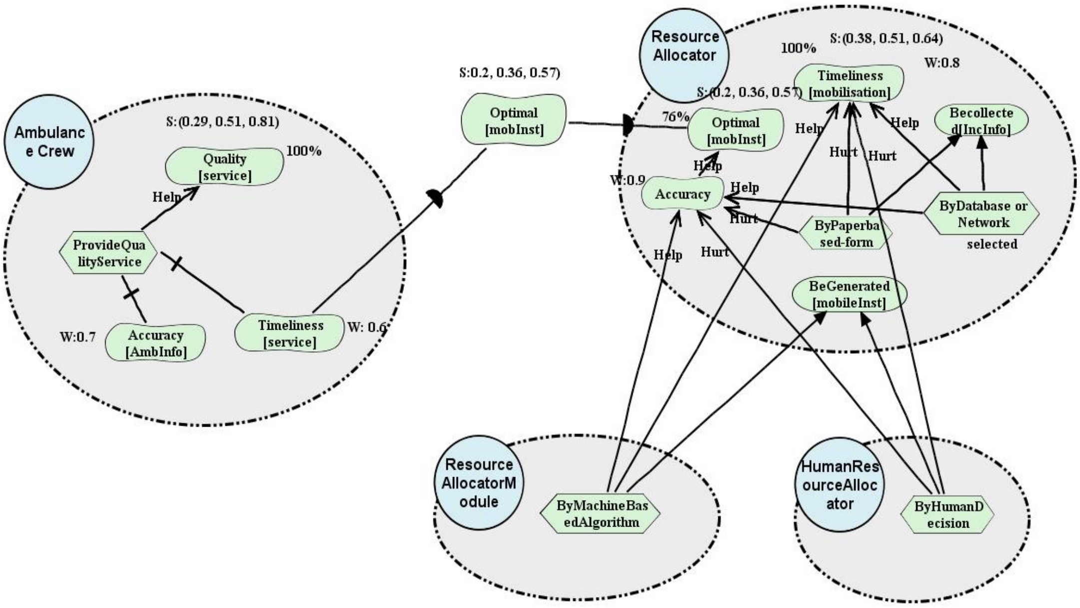

Fig. 3. Optimal soft goal scores for the alternative option *ByDatabase or Network.*

In regards to modelling, Sensitivity Analysis can help an analyst in numerous ways. Sensitivity analysis is one of the most appealing and interesting field in optimisation [25]. Efforts are made to explore the problem's behaviour for changes in the input data. The following questions are used by sensitivity analysis. What is the range of the input parameter? How positive or optimal are the results? How much will the result change if the data is slightly varied? Will these changes have a minor or a major impact on the results? Formally, the question is: Is optimal solution sensitive to a small change in one of the problem coefficients? Usually, variation occurs in the right hand side of the constraints and /or the objective function's coefficient. If the solution of the Linear Program (LP) changes, when the original coefficient is changed, then it is referred to as LP sensitive. Imagine the model of the linear form

$$Y = \sum_{i=1}^{n} C_i X_i$$

$$\text{lowerlimit} \le C_i \le \text{upperlimit}$$

where the input parameters are C_i, lowerlimit and upperlimit. Now by changing the values of the input parameters the sensitivity analysis is performed. A diagrammatic view for sensitive analysis for an optimisation model is given in Fig. 4. The Sensitivity analysis capability is provided by most optimisation tools; however data attained is dependent on continuous variables. This control on maintaining integer variables increases the complexity of the problem, and hence decreases the computational efficiency [26].

3.1. Implementation

To overcome the above mentioned issue, a simulation was created to check the system behaviour for change in each input parameter. The values of the input parameter are altered until a change in the solution takes place. The sensitive data provides the range for an input data of which there is no change in the optimal output value. The analyst is alerted if the value exceeds the range obtained from the sensitivity analysis. An analyst can take action by re-considering the input data. For our optimal *i** framework, the objective function is:

$$F = \text{Max} \sum_{i=1}^{m} \bar{\mathrm{C}} \times \omega$$

The input parameter on the right side of the objective function is the impact of the alternative on the leaf soft goals and it is given by triangular fuzzy number (a1, a2, a3). Usually the choice of fuzzy number varies from expert to expert. Hence, it is interesting to observe the dependency of the solutions obtained from the parameters of the fuzzy numbers. A special case is now considered in which these numbers are perturbed by δ1 and δ2 as in Fig. 5.

In this case, the task is to find the range in which δ1 and δ2 may vary without violating the optimal solution. The impacts of the goals or task to soft goals *make, help, some+, some-, hurt, and break* are represented by triangular fuzzy numbers (0.64, 0.80, 1), (0.48, 0.64, 0.80), (0.32, 0.48, 0.64), (0.16, 0.32, 0.48), (0, 0.16, 0.32) and (0, 0, 0.16) correspondingly. For each type of impact, the lower and upper bounds are varied, except *make* and *break,* because the impacts are represented as fuzzy numbers from 0 to 1. To improve the analysis, initially the lower (upper) bound is tested and moved on, only if the bound is not a limit for the analysis. The middle point between the current impact's lower (upper) limit, and adjacent impact's lower (upper) limit, forms the lower (upper) bound

$$m_l = (l_1+l_2)\ /2 \qquad m_u = (u_1+u_2)\ /2$$

where m_l is the middle point for lower limit, m_u is the middle point for the upper limit, l_1 is the current impact's lower limit, l_2 is the adjacent impact's lower limit, u_1 is the current impact's upper limit and u_2 is the adjacent impact's upper limit. The values are decreased (increased) from this middle point until a change in the optimal solution occurs.

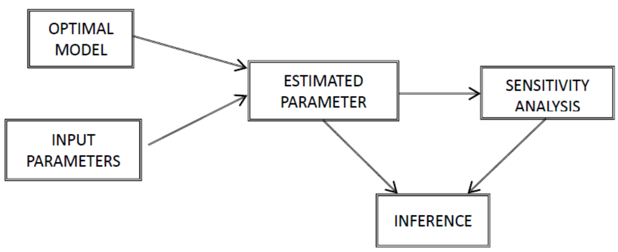

Fig. 4. A model for the sensitivity analysis of optimisation.

3.2. Analysis with LAS Case Study

Simulation was carried out for the case studies: Youth Counselling System [9], Meeting Scheduling System [11] and LAS [21]. These systems are chosen because of their simplicity and ease of comprehension. In this paper the sensitivity analysis for LAS is discussed. In the LAS goal model (Fig. 1), *help* and *hurt* are the two impacts associated with the alternative options. The fuzzy value of *help* and *hurt* is (0.48, 0.64, 0.8) and (0, 0.16, 0.32) respectively. The lower bound of *help* and upper bound of *hurt* are varied to check change in the optimal satisfaction levels of the top soft goals. The upper bound of *help* is not varied, due to the upper limit of the membership function being 1. Similarly, the lower bound of *hurt* is not changed, due to the lower limit of the membership function being 0. For example, for the actor *Resource Allocator,* the impact of the alternative option *ByDatabase or Network* on the leaf soft goal *Timeliness[mobilization]* has an impact value of (0.48, 0.64, 0.8). According to sensitivity analysis, the impact has the bounds {(0.48, 0.64, 0.8), (0.2, 0.64, 0.8)}; this implies that the impact can take any value in this range without a change in the optimal satisfaction levels of the soft goals. Similarly, the impact of the alternative option, *ByPaperbased Form* on the leaf soft goal *Timeliness[mobilization]* has an impact value of (0, 0.16, 0.32). According to sensitivity analysis, the impact was found to have the bounds {(0, 0.16, 0.32), (0, 0.16, 0.6)}.

Fig. 6 and Fig. 7 demonstrate the graphical representation of the sensitivity analysis for the top soft goals, *Timeliness[mobilization]* and *Optimal[mobInst]* of the actor *Resource Allocator,* with both impacts being taken into consideration*.* For instance the graph demonstrates the score of the soft goals for both of the impacts *help* and *hurt*. It is found that beyond the sensitivity analysis bounds of the impacts, the optimal satisfaction scores of the soft goals decrease and within the specified bound the optimal solution, remains unchanged.

The benefit of sensitivity analysis is that it helps the analyst when deciding if the inputs are within the accepted range. Also analysts can examine the solutions obtained from different inputs and decide upon the best solution. Furthermore, the analyst need not perform the sensitivity analysis every time optimisation is conducted; only when access to the data provided is required. This means that when an impact comes under review the analyst can make a demand for bound calculation.

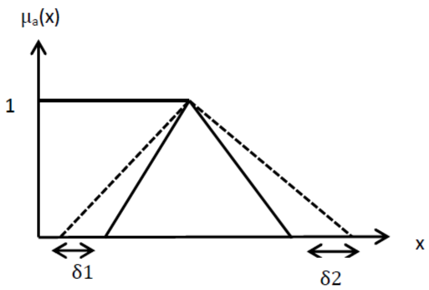

Fig. 5. Perturbation of fuzzy number.

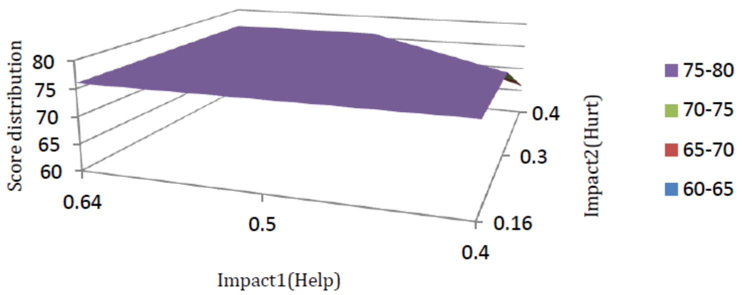

Fig. 6. Sensitivity analysis for *optimal soft goal.*

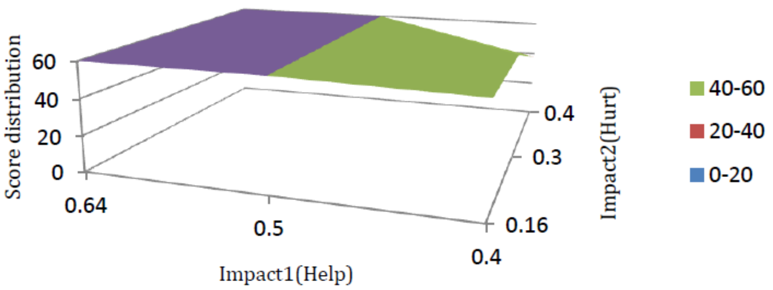

Fig. 7. Sensitivity Analysis for *Timeliness soft goal.*

4. Related Works

Since the development of the concept of the goal model, a considerable amount of work on the reasoning of goal achievement using qualitative and quantitative labels has been proposed. However, only a limited amount of research work has been done into the optimisation of the goal models. This section briefly describes the works related to some of the quantitative and qualitative goal analysis approaches as well as optimisation in goal analysis.

Lamsweerde [11] proposed a lightweight quantitative alternative analysis of goals in the KAOS framework to overcome the issues associated with qualitative analysis. In his approach he used variables such as gauge variable, ideal target value, and the maximum acceptable value associated with each soft goal. This approach obtained these values from the specification of the system. So to design a goal model using this method, one should have completely understood the specification of the system. Another problem with this approach is that it may be difficult to apply in the case of complex and large systems. D. Zowghi *et al*. [28] presented a Multi Criteria Decision Analysis (MCDA) for resolving the conflicts in NFR decision analysis. It used TOPSIS as a MCDA technique to rank the alternatives. It was done to assist the software developers in quantitative conflict decision-making analysis by integrating TOPSIS with their sure CM Framework. J. Mylopoulos *et al*. [29] presented a formal reasoning of goals in goal models. This is attained by presenting a qualitative formalization and label propagation algorithm. In addition, based on the probabilistic model, quantitative semantics for new relationships are given. This requires a strong mathematical knowledge, as it uses first order logic.

D.Amyot *et al*. [7] developed a hybrid approach by combining the qualitative and quantitative approaches to perform an analysis of the GRL model to evaluate the satisfaction levels of the actors and the intentional elements. Three algorithms, namely qualitative, quantitative and hybrid are implemented in the open-source jUCMNav tool, an Eclipse-based editor for URN models. The algorithms are illustrated using the example of a telecommunications system. Horkoff and Yu [9] proposed a qualitative analysis of goal models to comprehend the problem domain during an early phase of requirement engineering. In addition to comprehension of the problem domain, the model is used to perform elicitation, which requires customer intervention. However, the main issue with their approach is the ambiguity of the decision-making when one or more goals receive the same labels. Sidiq and Jain [30] presented a method for requirements prioritization, using an $\propto$-level weighted F-preference relation and a fuzzy based Analytical Hierarchal Process (AHP). The AHP pair wise comparison is used for assigning weights to goals/soft goals and locates the prioritized list of requirements using the binary sort tree method.

All the approaches discussed above in this section use qualitative reasoning, quantitative reasoning, probability and AHP to find the best possible solution for goal analysis. However, these approaches do not use optimisation, an operations research technique used to find optimal design option during analysis of goals. The research proposals that use optimisation in goal analysis are proposed by William *et al*. [15] and by Affleck *et al*. [19], [27]. William *et al*. [15] extended their previous work by developing a multi-objective optimisation model to the KAOS goal model for exploring the alternative design options. Their work was illustrated by using two case studies: the London Ambulance System (LAS) and financial fraud system. This approach uses probability distribution to simulate the vector values for each leaf quality variable. It does not evaluate the design options by considering impact of design decision on non-functional requirements. Affleck *et al*. [19], [27] has proposed a linear programming optimisation model to the NFR framework. This approach aims to minimise the operationalisations. To realise this goal, it employs single objective optimisation to select the minimum number of operations that maximises the overall satisfaction of non-functional requirements. Additionally, sensitivity analysis is performed to help developers find the quantitative set of input values.

Our work is one of the earliest attempt at applying quantitative and multi-objective optimisation model to the *i** framework. The proposed approach overcomes the problem of uncertainty that arise in decision making of the *i** qualitative analysis and thereby is an improvement over the existing interactive qualitative analysis of the *i** framework [9]. Even though William *et al*. and Affleck *et al*. have used optimisation, their approaches are different from ours. They applied optimisation to the KAOS and the NFR framework respectively, where as we have applied to the *i** framework. Affleck *et al*. used single objective optimisation function to the NFR framework but we have used multi-objective optimisation to the *i** framework. Although William *et al*. applied multi-objective optimisation on KAOS framework but they did not consider the impact of non-function requirements. On the contrary, this paper has proposed a multi-objective optimisation model of the *i** framework to evaluate the design options by considering impact of alternative designs on the non-functional requirements.

5. Conclusion

This paper has presented a method of representing *i** framework as a directed graph. This graph was then used to create a multi-objective function optimisation model for the *i** framework based on maximising the top soft goals of a given system. The optimisation model was used to obtain the weights of the leaf soft goals that are used in the goal analysis. The optimisation model was evaluated using the case studies from the existing literature, which discusses the London Ambulance System, the Youth Counsellor, the Meeting Scheduling System and the Telemedicine System. Furthermore, a sensitivity analysis approach was developed and implemented to examine the solution used for the London Ambulance System. The optimisation was helpful to the analyst in identifying the weights of leaf soft goals and thereby avoiding subjective selection. The sensitivity analysis aided the analyst to determine the bounds of the inputs, for which there is no change in the optimal solution. The future direction for this research is to develop a tool that can be used to implement multi-objective optimisation and sensitivity analysis. Furthermore, this tool would also be used to conduct an empirical validation to evaluate this proposal.

References

[1] Bashar N., & Steve, E. (2000). Requirements engineering: A roadmap. *Proceedings of the Conference on the Future of Software Engineering* (pp. 35-46).

[2] Glinz, M. (2007). *Proceedings of the 5th IEEE International Requirements Engineering Conference on Non-functional Requirements* (pp. 21–26).

[3] Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2000). *Non-functional Requirements in Software Engineering*. Kluwer Academic Publishers.

[4] Dardenne, A., Fickas, S. S., & Lamsweerde, A. V. (1991). Goal-directed concept acquisition in requirement elicitation. *Proceedings of the 6th International Workshop on Software Specification and*

Design (pp. 14-21).

[5] Yu, E. (1995). Modeling strategic relationships for process reengineering. PhD thesis, Department of Computer Science, University of Toronto, Toronto, Canada.

[6] Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (1995). Tropos: An agent oriented software development methodology. *Autonomous Agents and Multi-agents Systems*, 203-236.

[7] Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., & Yu, E. (2010). Evaluating goal models within the goal-oriented requirement language. *International Journal of Intelligent Systems, 25(8)*, 841-877.

[8] Kaiya, H., Horai, H., & Saeki, M. (2002). AGORA: Attributed goal-oriented requirements analysis method. *Proceedings of the 10th Anniversary IEEE Joint international Conference on Requirements Engineering,* (pp. 13–22).

[9] Horkoff, J., & E. Yu, (2009). Evaluating goal achievement in enterprise modeling – An interactive procedure and experiences. *Proceedings of the Conference on The Practice of Enterprise Modeling* (pp. 145-160).

[10] Affleck, A., & Krishna, A. (2012). Supporting quantitative reasoning of non-functional requirements: A process-oriented approach. *Software and System Process,* 88–92.

[11] Van, L. (2009). Reasoning about alternative requirements options. *Conceptual Modeling: Foundations and Applications*, 380–397.

[12] Chitra, M. S., Aneesh, K., Raj, P. G., & Arshinder, K. (2015). Quantitative reasoning of goal satisfaction in the *i** framework. *Proceedings of the 27th International Conference on Software Engineering and Knowledge Engineering.*

[13] Chitra, M. S., Aneesh, K., & Arshinder, K. (2015). Reasoning about goal satisfaction for early requirements engineering in the *i** framework using inter-actor dependency. *The 19th Pacific Asia Conference on Information Systems (PACIS 2015).* Singapore.

[14] Letier, E., & Lamsweerde, A. V. (2004). Reasoning about partial goal satisfaction for requirements and design engineering. *Proceedings of the 12th International Symposium on the Foundation of Software Engineering* (pp. 53– 62).

[15] Heaven, W., & Letier, E. (2011). Simulating and optimizing design decisions in goal models. *Proceedings of 19th IEEE International Requirements Engineering Conference*.

[16] Liaskos, S., McIlraith, S. A., Sohrabi, S., & Mylopoulos, J. (2010). Integrating preferences into goal models for requirements engineering. *Proceedings of the IEEE International Conference on Requirements Engineering* (pp 135–144).

[17] Johnson, L. A., & Montgomery, D. C. (1974). *Operations Research in Production Planning, Scheduling and Inventory Control.* John Wiley & Sons Inc.

[18] Sasieni, M., Yaspan, A., & Friedman, L. (1959). *Operations Research: Methods and Problems*. John Wiley & Sons Inc.

[19] Affleck, A., Krishna, A., & Achuthan, N. R. (2015). Non-functional requirements framework: A mathematical programming approach. *The Computer Journal, 58(5)*, 1122-1139.

[20] Yu, E., Giorgini, P., Maiden, N., & Mylopoulos, J. (2011). *Social Modeling for Requirements Engineering*. Cambridge, MA: MIT Press.

[21] You, Z. (2004). Using meta-model-driven views to address scalability in i* models. Master thesis, Department of Computer Science, University of Toronto, Canada.

[22] Harman, M. (2007). The current state and future of search based software engineering. *Future of Software Engineering*, 342-357.

[23] Caramia, D. (2008). *Multi-objective Management in Freight Logistics*. Springer-Verlag, London.

[24] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6(2)*, 182-197.

[25] Hadigheh, A. G., & Terlaky, T. (2006). Sensitivity analysis in linear optimization: Invariant support set intervals. *European Journal of Operational Research*, 1158–1175.

[26] lp solve. Introduction to lp_solve 5.5.2.0. Retrieved, from http://lpsolve.sourceforge.net

[27] Affleck, A., Krishna, A., & Achuthan, N. R. (2013). Optimal selection of operationalisations for non-functional requirements. *Proceedings of the 9th Asia-Pacific Conf. Conceptual Modelling* (pp. 69–78).

[28] Zowghi, D., *et al*. (2014). Utilizing topsis: A multi criteria decision analysis technique for non-functional requirements conflicts. *Requirements Engineering*.

[29] Giorgini, P., Mylopoulos, J., Nicchiarelli, E., & Sebastiani, R. (2003). Reasoning with goal models. *Lecture Notes in Computer Science*, 167–181.

[30] Mohd, S., & Jain, S. K. (2014). Applying fuzzy preference relation for requirements prioritation in goal oriented requirements elicitation process. *Int. J. Syste. Assur. Eng. Manag*, *5(4)*, 711-723.

Chitra M Subramanian is a PhD candidate in the Department of Computing at Curtin University, Perth, Western Australia. She has received her bachelor's in computer science and engineering from Madurai Kamaraj University, India and the masters in computer science and engineering from Anna University, India. Her research interests include software engineering, distributed systems and parallel computing.

Aneesh Krishna is currently senior lecturer of software engineering with the Department of Computing, Curtin University, Australia. He holds a PhD in computer science from the University of Wollongong, Australia, an M.Sc (Engg.) in electronics engineering from Aligarh Muslim University, India and a B.E. degree in electronics engineering from Bangalore University, India. He was a lecturer in software engineering at the School of Computer Science & Software Engineering, University of Wollongong, Australia (from February 2006 - June 2009). His research interests include software engineering, requirements engineering, conceptual modelling, agent systems, formal methods and service-oriented computing. His research is (or has been) funded by the Australian Research Council and various Australian government agencies as well as companies such as Woodside Energy, Amristar Solutions, Andrew Corporation, NSW State Emergency Service, Western Australia Dementia Study Centre and Autism West. He serves as an assessor (Ozreader) for the Australian Research Council. He has been on the organising committee, served as invited technical program committee member of many conferences and workshop in the areas related to his research.

Arshinder Kaur is currently an associate professor in the Department of Management Studies, Indian Institute of Technology, Madras, India. She did her PhD from IIT Delhi, India in the area of supply chain management. Her doctoral work (PhD thesis) has been accepted as a Highly Commended Award Winner of the 2008 Emerald/EFMD Outstanding Doctoral Research Award. She has many publications in reputed international journals like Int. J. of Production Economics, Production Planning and Control, Computers and Industrial Engineering and Int. J. of Logistics and Systems Management. Her research interests are supply chain contracts, supply chain coordination, inventory management, closed loop supply chain, service operations and strategic sourcing.